AWS 白皮書

AWS Lambda 安全概觀

AWS Lambda 安全概觀: AWS 白皮書

Table of Contents

摘要 i
摘要 1
簡介 2
關於 AWS Lambda 3
Lambda 的優勢 3
不需管理伺服器 3
持續擴展 4
毫秒級計量 4
增加創新 4
現代化應用程式 4
豐富的生態系統 4
執行 Lambda 式應用程式的成本 4
AWS 共同責任模式 5
Lambda 函數 6
Lambda 叫用模式 7
Lambda 執行 9
Lambda 執行環境 9
執行角色 10
Lambda MicroVM 與工作者 10
Lambda 隔離技術 12
儲存和狀態 12
Lambda 中的執行時間維護 14
監控和稽核 Lambda 函數 15
Amazon CloudWatch 15
Amazon CloudTrail 15
AWS X-Ray 15
AWS Config 15
架構及操作 Lambda 函數 16
Lambda 與合規 17
Lambda 事件來源 18
結論 19
作者群 20
深入閱讀 21
文件修訂 22

聲明 23

AWS Lambda 安全概觀

出版日期：2021 年 2 月 12 日 (文件修訂)

摘要

本白皮書透過安全焦點深入介紹 AWS Lambda 服務。其提供了服務的全面概觀，不僅對新的採用者而言相當實用，也能加深現行使用者對 Lambda 的理解。

本白皮書的目標讀者是首席資訊安全長 (CISO)、資訊安全工程師、企業架構師、合規團隊，以及任何有興趣了解 AWS Lambda 基礎技術的人員。

簡介

如今有越來越多的工作負載使用 AWS Lambda 以實現可擴展性、效能及成本效益，且無需管理底層的基礎設施。這些工作負載可擴展至每秒上千個並行請求。Lambda 是 AWS 目前所提供許多重要服務的其中一項。成千上萬的 Amazon Web Services (AWS) 客戶每個月都使用 Lambda 來處理數億個請求。

Lambda 適合許多業界的任務關鍵型應用程式。從媒體及娛樂業，到金融服務及其他受法規限制的產業，各式各樣的客戶都充分利用了 Lambda。這些客戶專注在其最擅長的任務：進行業務，不僅減少了上市所需的時間、將成本最佳化，還改善了敏捷性。

受管的執行階段環境模型，使 Lambda 得以管理執行無伺服器工作負載的許多實作細節。這種模型在更進一步減少受攻擊面的同時，也使得雲端安全更加簡單。本白皮書描述了這種模型的基礎技術，以及適用於開發人員、安全分析師、安全及合規團隊，以及其他利害關係人的最佳實務。

關於 AWS Lambda

AWS Lambda 是一種事件驅動的無伺服器運算服務，可透過自訂邏輯擴充其他 AWS 服務，或是建立其他可大規模運作並兼具效能及安全的後端服務。Lambda 可自動執行程式碼來回應多個事件，例如透過 Amazon API Gateway 發出 HTTP 請求、修改 Amazon S3 儲存貯體中的物件、更新 Amazon DynamoDB 中的資料表，以及在 AWS Step Functions 中進行狀態轉換。您還可以直接從任何 Web 或行動應用程式執行程式碼。Lambda 會在高度可用的運算基礎設施上執行程式碼，並執行所有基礎平台的管理，包括伺服器和作業系統維護，容量佈建與自動擴展、修補、程式碼監控及記錄日誌。

有了 Lambda，您只要上傳程式碼並設定何時叫用；Lambda 會負責所有其他工作，在高可用性的情況下執行您的程式碼。Lambda 與許多其他 AWS 服務整合，可讓您建立無伺服器應用程式或後端服務，範圍從定期觸發的簡單自動化任務到完整的微服務應用程式都有。

Lambda 也能進行設定，以存取您 Amazon Virtual Private Cloud 內的資源，並透過擴充套件存取您的內部部署資源。

您可以使用 AWS Identity and Access Management (IAM) 輕易地為 Lambda 加上強而有力的安全狀態，並使用其他本白皮書中討論的技術來維持高水準的安全及稽核，以滿足您的合規需求。

主題

- Lambda 的優勢
- 執行 Lambda 式應用程式的成本

Lambda 的優勢

希望發揮創造力及提高開發組織速度，同時不想犧牲 IT 團隊提供可擴展、符合成本效益及可管理基礎設施能力的客戶，會發現 AWS Lambda 可使其以操作複雜性換取敏捷性及更佳的定價，同時不必犧牲規模或可靠性。

Lambda 提供許多優勢，其他包括下列幾項：

不需管理伺服器

Lambda 會在橫跨單一區域中多個可用區域 (AZ)、高度可用且容錯的基礎設施上執行您的程式碼，無縫部署程式碼，並提供所有基礎設施的管理、維護及修補。Lambda 也提供內建的記錄日誌和監控，包括與 Amazon CloudWatch、CloudWatch Logs 以及 AWS CloudTrail 的整合。

持續擴展

Lambda 會透過以平行方式執行事件觸發的程式碼，並個別處理每個事件，以精確地管理您函數 (或應用程式) 的擴展。

毫秒級計量

使用 AWS Lambda 時，是以您程式碼的執行時間 (按每 1 毫秒 (ms) 計算)，以及您程式碼的觸發次數向您收取費用。您會依穩定的輸送量或執行的持續時間支付費用，而非伺服器單位。

增加創新

Lambda 會接管基礎設施管理，以此釋放您的程式設計資源，讓團隊可以更專注在創新及開發商業邏輯。

現代化應用程式

Lambda 可讓您使用函數與預先訓練的機器學習模型，輕鬆地將人工智慧注入到應用程式中。單一應用程式介面 (API) 請求可分類影像、分析影片、將語音轉換成文字、執行自然語言處理等工作。

豐富的生態系統

Lambda 可透過各種項目支援開發人員，其中包括 AWS Serverless Application Repository (用於探索、部署及發佈無伺服器應用程式)、AWS Serverless Application Model (用於建置無伺服器應用程式及與各種整合開發環境 (IDE) 整合，例如 AWS Cloud9、AWS Toolkit for Visual Studio、AWS Tools for Visual Studio Team Services 及其他幾種)。Lambda 還與其他 AWS 服務整合，可為您提供豐富的生態系統以建置無伺服器應用程式。

執行 Lambda 式應用程式的成本

Lambda 提供精細的按使用量付費定價模型。透過此模型，您會根據函數的叫用次數及叫用的持續時間 (執行程式碼所耗費的時間) 支付費用。除了這種靈活的定價方式外，Lambda 也提供了每個月 100 萬次的永久免費請求，可讓許多客戶在無需負擔任何成本的情況下自動化其程序。

AWS 共同責任模式

安全與合規是 AWS 和客戶的共同責任。這種共同責任模式有助於減輕您的營運負擔，因為 AWS 會深入服務運作所在設施的實體安全，操作、管理並控制主機作業系統及虛擬化層的元件。

針對 AWS Lambda，AWS 會管理底層基礎設施和基礎服務、作業系統和應用程式平台。您需負責您程式碼的安全，以及對 Lambda 服務及您函數中的 Identity and Access Management (IAM)。

圖 1 顯示適用於通用和不同 AWS Lambda 元件的共同責任模式。AWS 責任以橙色顯示在虛線下方，客戶責任則以藍色顯示在虛線上方。

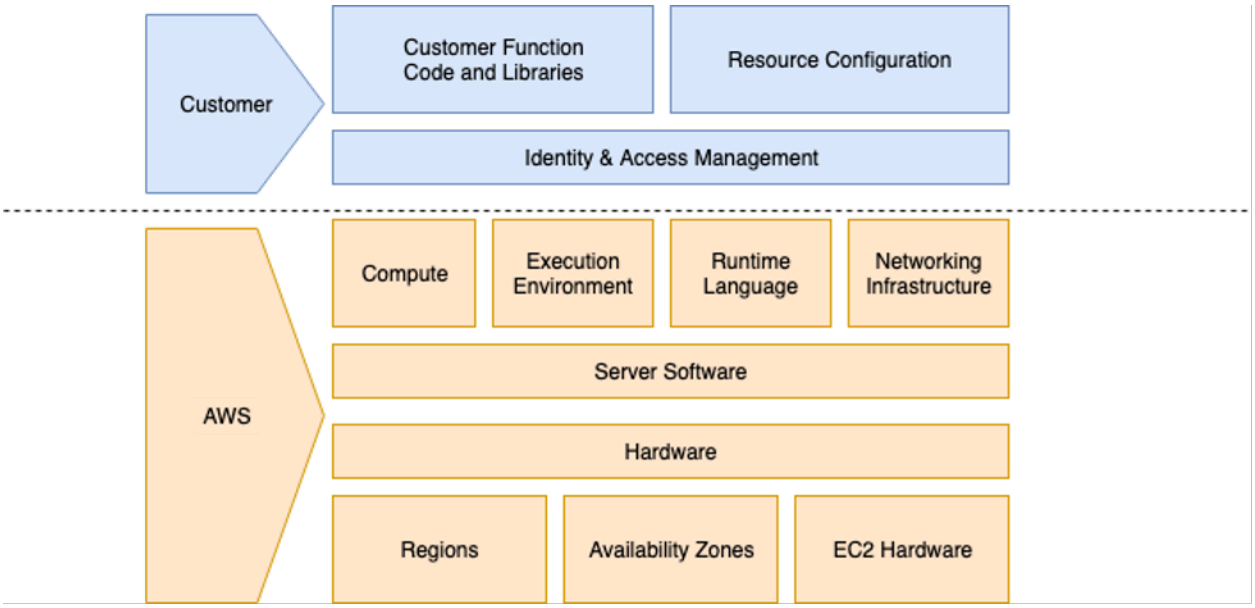

圖 1 — AWS Lambda 的共同責任模式

Lambda 函數與層

使用 Lambda，您可以在幾乎完全無需管理底層基礎設施的情況下執行程式碼。您僅需負責為 Lambda 提供程式碼，以及設定 Lambda 代您執行程式碼的方式。目前，Lambda 支援兩種程式碼資源類型：函數及層。

函數是一種可進行叫用，以在 Lambda 函數中執行您程式碼的資源，其中可包括稱為層的通用或共享資源。層可以用來在不同的函數或 AWS 帳戶間共享通用程式碼或資料。您負責管理所有包含在您函數或層內的程式碼。當 Lambda 收到來自客戶的函數或層程式碼時，Lambda 會透過使用 AWS Key Management Service (AWS KMS) 對其進行靜態加密，以及使用 TLS 1.2+ 對其進行傳輸中加密，以保護其存取。

您可以透過 AWS Lambda 政策，或透過資源型許可管理對您函數或層的存取。如需 IAM 上支援的 IAM 功能完整清單，請參閱可和 IAM 配合使用的 AWS 服務。

您也可以透過 Lambda 的控制平面 API 管理函數和層的整個生命週期。例如，您可以選擇呼叫 `DeleteFunction` 來刪除您的函數，或是呼叫 `RemovePermission` 來撤銷另一個帳戶的許可。

Lambda 叫用模式

叫用 API 可以透過兩種模式呼叫：事件模式和請求回應模式。

- 事件模式會將酬載排入佇列，以進行非同步叫用。
- 請求回應模式會使用提供的酬載，以同步方式叫用函數，並立即傳回回應。

在這兩種情況下，函數執行都一律會在 Lambda 執行環境中進行，但酬載會採取不同的路徑。如需詳細資訊，請參閱本文件中的〈Lambda 執行環境〉。

您也可以使用其他代您執行叫用的 AWS 服務。要使用哪一種叫用模式，取決於您正在使用的 AWS 服務，以及其設定方式。如需其他 AWS 服務與 Lambda 整合方式的額外資訊，請參閱搭配其他服務使用 AWS Lambda。

當 Lambda 收到請求回應叫用時，會直接將該請求傳遞給叫用服務。如果叫用服務無法使用，呼叫端可能會暫時將酬載用戶端排入佇列，以重試叫用一定次數。若叫用服務收到酬載，服務接著會嘗試識別可用於該請求的執行環境，並將酬載傳遞給該執行環境以完成叫用。如果不存在現有或適合的執行環境，則會動態建立一個執行環境以回應請求。在傳輸途中，傳送到叫用服務的叫用酬載會受到 TLS 1.2+ 保護。Lambda 服務內的流量 (自負載平衡器以下) 會經過隔離的內部 virtual private cloud (VPC)，這個 virtual private cloud 在傳送請求的 AWS 區域內會由 Lambda 服務所擁有。

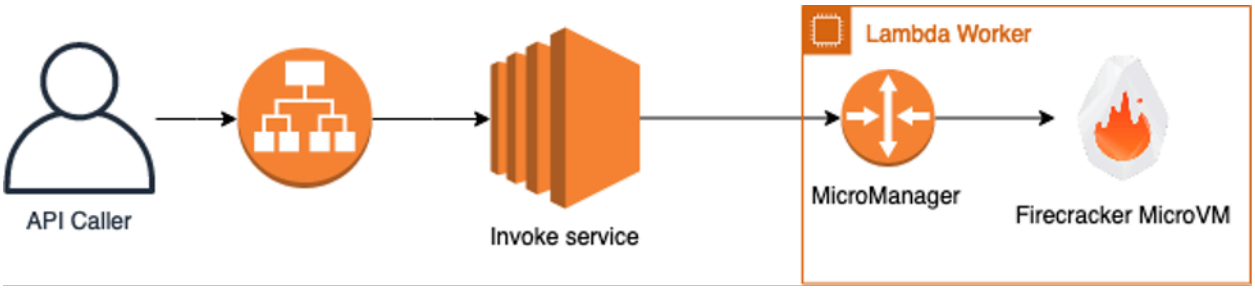

圖 2 – AWS Lambda 請求回應叫用模型

事件叫用模式酬載一律會在叫用前排入佇列以進行處理。所有酬載都會排入佇列，以在 Amazon Simple Queue Service (Amazon SQS) 佇列中進行處理。排入佇列的事件一律會在傳輸中受到 TLS 1.2+ 保護，但目前並未受到靜態加密。由於 Lambda 使用的 Amazon SQS 佇列是由 Lambda 服務所管理，因此您做為客戶無法看見。排入佇列的事件可以存放在共享的佇列中，但取決於各種無法由客戶直接控制的因素 (例如叫用率、事件大小等)，可能會遷移或指派給專用的佇列。

排入佇列的事件會由 Lambda 的輪詢器機群以批次方式擷取。輪詢器機群是一組 EC2 執行個體，其目的是處理已排入佇列但尚未處理的事件叫用。當輪詢器機群擷取已排入佇列且需要處理的事件時，會採用與客戶在請求回應模式叫用中所採用的相同方式，將事件傳遞給叫用服務。

若無法執行叫用，輪詢器機群將會暫時將事件存放在主機的記憶體中，直到可以成功完成執行，或是直到超過執行重試的嘗試次數為止。酬載資料永遠不會寫入輪詢器機群本身的磁碟。輪詢器機群可以跨 AWS 客戶分配任務，以取得最短的叫用時間。如需哪些服務可能會採取事件叫用模式的詳細資訊，請參閱搭配其他服務使用 AWS Lambda。

Lambda 執行

當 Lambda 代您執行函數時，會同時管理佈建及設定執行程式碼所需的基礎系統。這可讓您的開發人員專注在商業邏輯及撰寫程式碼，而非管理基礎系統。

Lambda 服務分成控制平面和資料平面。每個平面都在服務中提供不同的用途。控制平面提供管理 API (例如 `CreateFunction`、`UpdateFunctionCode`、`PublishLayerVersion` 等)，並會管理與所有 AWS 服務的整合。與 Lambda 控制平面的通訊在傳輸中會受到 TLS 保護。所有存放在 Lambda 控制平面內的客戶資料都會透過使用 AWS KMS 進行靜態加密，其設計旨在保護資料不受未經授權的洩露或竄改。

資料平面是 Lambda 的叫用 API，會在叫用 Lambda 函數時觸發。叫用 Lambda 函數時，資料平面會在 AWS Lambda 工作者 (或直接稱呼為工作者，這是一種 Amazon EC2 執行個體類型) 上配置執行環境給該函數版本，或是選擇已為該函數版本設定的現有執行環境，然後使用該環境完成叫用。如需詳細資訊，請參閱本文件中的〈AWS Lambda MicroVM 與工作者〉一節。

Lambda 執行環境

每一次的叫用都會由 Lambda 叫用服務路由至工作者上能夠處理該請求的執行環境。除了透過資料平面外，客戶和其他使用者都無法直接啟動與執行環境的傳入/輸入網路通訊。這有助於確保與您執行環境的通訊經過身分驗證及授權。

執行環境會針對特定函數版本進行保留，並且無法在不同函數版本、不同函數或 AWS 帳戶間重複使用。這表示單一函數可能會具有兩個不同版本，並可能會導致至少兩個獨特的執行環境。

每個執行環境一次只能用於一個並行叫用，並且基於效能考量，執行環境可以在相同函數版本的多次叫用間重複使用。取決於許多因素 (例如叫用率、函數組態等)，特定函數版本可能會有一或多個執行環境。透過這種方式，Lambda 能夠為其客戶提供函數版本層級的隔離。

Lambda 目前不會隔離函數版本執行環境內的叫用。這表示其中一次叫用可能會留下一個狀態，而該狀態可能會影響下一次叫用 (例如寫入 /tmp 的檔案或記憶體中的資料)。如果您希望確保其中一次叫用不會影響另外一次叫用，Lambda 建議您建立其他不同的函數。例如，您可以為更容易發生錯誤的複雜剖析作業建立不同函數，並重複使用不會執行安全敏感作業的函數。Lambda 目前沒有限制客戶能建立的函數數量。如需限制的詳細資訊，請參閱 Lambda 配額頁面。

執行環境會持續受到 Lambda 的監控及管理，並且可能會因為各種理由而建立或刪除，這些理由包括但不限於：

- 新的叫用抵達，且不存在適合的執行環境

- 發生內部執行時間部署或工作者軟體部署
- 發佈了新的佈建並行執行組態
- 執行環境或工作者的租用時間正逐漸接近，或是已超過最大生命週期
- 其他的內部工作負載重新平衡程序

客戶可以在其函數組態上設定佈建並行執行，以管理為某個函數版本存在的預先佈建執行環境數量。當被設定執行這項作業時，Lambda 將會建立、管理及確保一律存在的設定執行環境數。這可確保客戶可對其任何規模的無伺服器應用程式啟動效能進行更大程度的控制。

除了透過佈建並行執行組態外，客戶無法決定性地控制 Lambda 為回應叫用而建立或管理的執行環境數。

執行角色

每個 Lambda 函數也必須設定執行角色，這是一種 IAM 角色，Lambda 服務會在執行與函數相關的控制平面和資料平面作業時取得這個角色。Lambda 服務會取得這個角色以擷取暫時安全憑證，這項安全憑證會在函數的叫用期間做為環境變數提供使用。基於效能考量，Lambda 服務會快取這些憑證，並且可能會在使用相同執行角色的不同執行環境之間重複使用這些憑證。

為了確保遵守最低權限原則，Lambda 建議每個函數都具備自身的獨特角色，並且使用其需要的最小許可設定該角色。

Lambda 服務也可能取得執行角色以執行特定控制平面作業，例如與建立和設定 VPC 函數彈性網路界面 (ENI) 有關的作業、傳送日誌到 Amazon CloudWatch Application Insights 的作業、傳送追蹤到 AWS X-Ray 的作業，或是其他與叫用無關的作業。客戶可以在 AWS CloudTrail 中檢閱稽核日誌，隨時檢閱和稽核這些使用案例。

如需此主題的詳細資訊，請參閱 AWS Lambda 執行角色文件頁面。

Lambda MicroVM 與工作者

Lambda 會在稱為 AWS Lambda 工作者的 Amazon EC2 執行個體機群上建立執行環境。工作者是裸機的 EC2 Nitro 執行個體，會由 Lambda 在不同、隔離，且客戶無法看見的 AWS 帳戶中啟動及管理。工作者具備由 Firecracker 建立的一或多個硬體虛擬化微型虛擬機器 (MVM)。Firecracker 是一種開放原始碼的虛擬機器監控器 (VMM)，使用以 Linux 核心為基礎的虛擬機器 (KVM) 建立及管理 MVM。其專為建立和管理安全、多租用戶的容器和函數型服務所建置，這些容器和服務會提供無伺服器作業模型。如需 Firecracker 安全模型的詳細資訊，請參閱 Firecracker 專案網站。

做為共同責任模式的一部分，Lambda 會負責維護安全組態、控制，以及修補工作者層級。Lambda 團隊使用 Amazon Inspector 探索已知的潛在安全問題，以及其他自訂安全問題通知機制和預先揭露的清單，使客戶無需管理其執行環境的基礎安全狀態。

圖 3 – AWS Lambda 工作者的隔離模型

工作者的最大租用生命週期為 14 小時。當工作者接近最大租用時間時，不會再有任何叫用路由至該工作者；MVM 會正常終止，並且基礎工作者執行個體也會終止。Lambda 會持續監控和針對其機群生命週期的生命週期活動進行警示。

所有與工作者進行的資料平面通訊都會使用 Galois/計數器模式的進階加密標準 (AES-GCM) 加密。由於工作者是託管在 Lambda 服務帳戶中由 Lambda 所管理的網路隔離 Amazon VPC 內，因此除了通過資料平面的作業外，客戶無法直接與工作者互動。

當工作者需要建立新的執行環境時，其會獲得具備時間限制的授權以存取客戶函數成品。這些成品專為 Lambda 的執行環境和工作者進行了最佳化。使用 ZIP 格式上傳的函數程式碼會先進行一次最佳化，然後會使用 AWS 管理的金鑰和 AES-GCM，以加密格式存放。

使用容器映像格式上傳到 Lambda 的函數也會進行最佳化。首先會從原始來源下載容器映像，最佳化為不同區塊，然後使用經過身分驗證的收斂加密方法存放為加密區塊，這種加密方法會使用 AES-CTR、AES-GCM 和 SHA-256 MAC 的組合。收斂加密方法可讓 Lambda 安全地消除加密區塊的重複資料。解密客戶資料所需的所有金鑰均會使用客戶管理的 AWS KMS 客戶主金鑰 (CMK) 進行保護。Lambda 服務的 CMK 用量可在 AWS CloudTrail 日誌中供客戶使用，以進行追蹤和稽核。

Lambda 隔離技術

Lambda 使用各種開放原始碼和專有隔離技術來保護工作者和執行環境。每個執行環境都包含下列項目的專用複本：

- 特定函數版本的程式碼
- 為您的函數版本選取的任何 AWS Lambda 層
- 選擇的函數執行時間 (例如 Java 11、NodeJS 12、Python 3.8 等) 或函數的自訂執行時間
- 可寫入的 /tmp 目錄
- 基於 Amazon Linux 2 的最小 Linux 使用者空間。

執行環境會使用數種內建於 Linux 核心的容器類技術，以及 AWS 專有的隔離技術，與其他執行環境隔離。這些技術包括：

- cgroups – 用來限制函數對 CPU 和記憶體的存取。
- namespaces – 每個執行環境都會在專用的命名空間內執行。我們會透過具備唯一的群組程序 ID、使用者 ID、網路界面及其他由 Linux 核心管理的資源進行這項作業。
- seccomp-bpf – 限制執行環境內所能使用的系統呼叫 (syscall)。
- iptables 和路由表 – 防止輸入網路通訊及隔離 MVM 之間的網路連線。
- chroot – 為基礎檔案系統提供範圍存取。
- Firecracker 組態 – 用來對區塊型儲存設備及網路裝置輸送量進行速率限制。
- Firecracker 安全功能 – 如需 Firecracker 目前安全設計的詳細資訊，請參閱 Firecracker 最新的設計文件。

搭配 AWS 專有的隔離技術，這些機制可在執行環境間提供強大的隔離。

儲存和狀態

執行環境永遠不會在不同函數版本或客戶間重複使用，但單一環境可能會在相同函數版本的叫用間重複使用。這表示資料和狀態可能會在叫用間持續存在。資料和/或狀態可能會持續存在數小時，才會做為一般執行環境生命週期管理的一部分遭到刪除。基於效能考量，函數可以透過保存和重複使用本機快取或叫用之間長時間執行的連線，利用這項行為來改善效率。在執行環境中，這些多個叫用會由單一程序處理，因此任何涉及整個程序的狀態 (例如 Java 中的靜態狀態) 都可供未來的叫用重複使用 (若叫用在重複使用的執行環境中發生的話)。

每個 Lambda 執行環境也會包含可寫入的檔案系統，可在 /tmp 取得。這個儲存體無法跨執行環境存取或共享。與程序狀態相同，寫入 /tmp 的檔案會在執行環境的生命週期期間保存。這可以讓您將昂貴的傳輸作業 (例如下載機器學習 (ML) 模型) 分攤到多個叫用中。不希望在叫用間保存資料的函數不應寫入 /tmp，或是必須在叫用間刪除其位於 /tmp 中的檔案。/tmp 目錄的後端是 Amazon EC2 執行個體存放區，並且會受到靜態加密。

希望將資料保存到執行環境外部檔案系統的客戶應考慮使用 Lambda 與 Amazon Elastic File System (Amazon EFS) 的整合。如需詳細資訊，請參閱搭配 AWS Lambda 使用 Amazon EFS。

若客戶不希望在叫用間保存資料或狀態，Lambda 建議客戶不要使用執行內容或執行環境來存放資料或狀態。如果客戶希望主動防止資料或狀態在叫用間遭到洩露，Lambda 建議客戶為每個狀態建立不同的函數。Lambda 不建議客戶使用或將安全敏感狀態存放到執行環境，因為其可能會在叫用間遭到變更。我們建議改為針對每次叫用重新計算狀態。

Lambda 中的執行時間維護

Lambda 透過持續掃瞄和部署相容的更新及安全修補程式，以及執行其他執行時間維護活動，為這些執行時間提供支援。這讓客戶得以專注在任何包含在其函數和層中的程式碼維護及安全。Lambda 團隊使用 Amazon Inspector 探索已知的安全問題，以及其他自訂安全問題通知機制和預先揭露的清單，確保我們的執行時間語言和執行環境維持在已修補的狀態。若發現任何新的修補程式或更新，Lambda 會測試和部署執行時間更新，且無需客戶介入。如需 Lambda 合規計劃的詳細資訊，請參閱本文件中的〈Lambda 與合規〉一節。

通常無需採取任何動作就可為支援的 Lambda 執行時間取得最新的修補程式，但有時候可能仍需要採取動作，以在部署前測試修補程式 (例如已知的不相容執行時間修補程式)。若需要客戶採取任何動作，Lambda 將會透過 Personal Health Dashboard、AWS 帳戶的電子郵件，或是透過其他方式聯絡客戶，並提供需要採取的特定動作。

客戶可以透過實作自訂執行時間，在 Lambda 中使用其他程式設計語言。針對自訂執行時間，客戶需要負責維護執行時間，包括確保自訂執行時間包括最新的安全修補程式。如需詳細資訊，請參閱《AWS Lambda 開發人員指南》中的自訂 AWS Lambda 執行時間。

當上游執行時間語言維護人員將其語言標記為生命週期結束 (EOL) 時，Lambda 不再支援該執行時間語言版，以遵照這一點。當執行時間版本在 Lambda 中標記為已遭棄用時，Lambda 會停止支援建立新函數及停止支援更新使用遭棄用執行時間編寫的現有函數。為了提醒客戶即將到來的執行時間棄用，Lambda 會傳送通知給客戶，並讓客戶了解即將到來的棄用日期及應預期的狀況。Lambda 也不會為棄用的執行時間提供安全更新、技術支援或修補程序，且保留隨時停止叫用設定為在遭棄用執行時間上執行函數的權利。若客戶希望繼續執行遭棄用或不支援的執行時間版本，可以自行建立自訂 AWS Lambda 執行時間。如需執行時間何時會遭到棄用的詳細資訊，請參閱 AWS Lambda 執行時間支援政策。

監控和稽核 Lambda 函數

您可以使用許多 AWS 服務和方法來監控和稽核 Lambda 函數，包括下列服務。

Amazon CloudWatch

AWS Lambda 會自動代您監控 Lambda 函數。透過 Amazon CloudWatch，其會報告請求數、每個請求的執行持續時間，以及導致錯誤的請求數等指標。這些指標會在函數層級公開，讓您可以利用這些指標來設定 CloudWatch 警示。如需 Lambda 公開的指標清單，請參閱 AWS Lambda 指標。

Amazon CloudTrail

透過使用 AWS CloudTrail，您可以針對整個 AWS 帳戶實作管控、合規、營運稽核和風險稽核，包括 Lambda。CloudTrail 可讓您記錄日誌、持續監控，並且保存與您整個 AWS 基礎設施內動作相關的帳戶活動，提供透過 AWS Management Console、AWS 開發套件、命令列工具及其他 AWS 服務所採取動作的完整事件歷史記錄。您可以利用 CloudTrail 選擇使用 AWS KMS 加密日誌檔案，同時利用 CloudTrail 日誌檔案完整性驗證以進行正向聲明。

AWS X-Ray

您可以使用 AWS X-Ray 分析生產及分散式 Lambda 型應用程式，並對其加以進行偵錯，了解應用程式的效能以及其基礎服務，進而最終識別和針對效能問題和錯誤的根本原因進行故障診斷。在通過您的應用程式時，X-Ray 的端對端請求檢視會顯示應用程式基礎元件的地圖，讓您可以在開發和生產期間分析應用程式。

AWS Config

透過 AWS Config，您可以追蹤 Lambda 函數的組態變更 (包括遭到刪除的函數)、執行階段環境、標籤、處理常式名稱、程式碼大小、記憶體配置、逾時設定和並行執行設定，以及 Lambda IAM 執行角色、子網路和安全群組關聯。這可以為您提供 Lambda 函數生命週期的全方位檢視，讓您顯示該資料以滿足潛在的稽核和合規需求。

架構及操作 Lambda 函數

本節討論 Lambda 架構及操作。如需無伺服器應用程式標準最佳實務的相關資訊，請參閱《無伺服器應用程式焦點》白皮書，其定義及探索了無伺服器情境下 AWS Well-Architected Framework 的支柱。

- 卓越營運支柱 – 透過執行和監控系統實現商業價值，以及持續改善支援流程與程序的能力。
- 安全支柱 – 保護資訊、系統和資產，同時透過風險評定和緩解策略提供商業價值的能力。
- 可靠性支柱 – 包括從基礎設施或服務中斷復原、動態取得運算資源以符合需求，以及緩解中斷狀況 (例如設定錯誤或暫時性網路問題) 的系統能力。
- 效能達成效率支柱 – 包括有效率地使用運算資源以滿足需求，並隨著需求變更與技術發展來維持該效率需求的能力。
- 成本最佳化支柱 – 持續改進及改善，在需求變更與技術發展的同時，將成本降至最低，同時確保實現業務成果的流程。

《無伺服器應用程式焦點》白皮書包括了各項主題，其中包含記錄指標和警示、調節和限制、為 Lambda 函數指派許可，以及將敏感資料提供給 Lambda 函數使用等主題。

Lambda 與合規

如〈共同責任模式〉一節所述，您需負責判斷哪些合規制度適用於您的資料。在您判斷合規制度需求後，您可以使用各種 Lambda 功能來滿足這些控制。您可以聯絡 AWS 專家 (例如解決方案架構師、網域專家、技術客戶經理及其他人力資源) 以取得協助。但是，AWS 無法建議客戶合規制度是否適用，或是哪些合規制度適用於特定的使用案例。

截至 2020 年 11 月，Lambda 屬於 SOC 1、SOC 2 及 SOC 3 報告的範圍中，這些報告是獨立的第三方檢驗報告，其中展現了 AWS 如何實現關鍵合規控制與目標。如需合規資訊的最新資訊清單，請參閱合規計劃的 AWS 服務範圍頁面。

由於一些合規報告的敏感性，這些報告無法公開共享。若要存取這些報告，您可以登入 AWS Management Console，然後使用 AWS Artifact，這是一種無成本的自助服務入口網站，可用來隨需存取 AWS 合規報告。

Lambda 事件來源

Lambda 透過直接整合，與超過 140 項 AWS 服務整合，並且也與 Amazon EventBridge 事件匯流排整合。常用的 Lambda 事件來源包括：

- Amazon API Gateway
- Amazon CloudWatch Events
- Amazon CloudWatch Logs
- Amazon DynamoDB Streams
- Amazon EventBridge
- Amazon Kinesis Data Streams
- Amazon S3
- Amazon SNS
- Amazon SQS
- AWS Step Functions

透過這些事件來源，您可以：

- 使用 AWS Identity and Access Management 安全地管理對服務和資源的存取。

- 靜態加密您的資料。* 所有服務都會對資料進行傳輸中加密。
- 使用 VPC 端點 (使用 AWS PrivateLink)，從您的 Amazon Virtual Private Cloud 進行存取。
- 使用 Amazon CloudWatch Application Insights 收集、報告及針對指標進行警示。
- 使用 AWS CloudTrail 記錄日誌、持續監控，並且保存與您整個 AWS 基礎設施內動作相關的帳戶活動，提供透過 AWS Management Console >AWS 開發套件、命令列工具及其他 AWS 服務所採取動作的完整事件歷史記錄。

* 在本白皮書出版時，Amazon EventBridge 尚不提供靜態加密資料。請繼續監控服務首頁，以了解這些功能的更新。

結論

AWS Lambda 提供了強大的工具組，可用來建置安全且可擴展的應用程式。Lambda 中的許多安全和合規最佳實務都與所有 AWS 服務相同，但有些為 Lambda 特有。本白皮書介紹了 Lambda 的優勢、其對應用程式的適用性，以及由 Lambda 管理的執行階段環境。本白皮書也包括了監控及稽核的詳細資訊，以及安全和合規性最佳實務。在您思考下一個實作時，請考慮您對 AWS Lambda 的理解，以及其可能可以透過何種方式改善您的工作負載解決方案。

作者群

此文件的作者包括：

- Global Life Sciences 解決方案架構師 Mayank Thakkar
- 資深首席工程師 (無伺服器) Marc Brooker
- 資深安全工程師 (無伺服器) Osman Surkatty

深入閱讀

如需其他資訊，請參閱：

- 共同責任模式，其說明了 AWS 一般而言如何看待安全。
- AWS 安全最佳實務涵蓋了適用於 AWS Identity and Access Management (IAM) 服務的建議。
- 《無伺服器應用程式焦點》涵蓋了 AWS Well-Architected Framework 並識別了關鍵要素，以確保您的工作負載架構符合最佳實務。
- 《AWS 安全簡介》提供了在 AWS 中思考安全的廣泛簡介。
- 《AWS 風險與合規》提供了 AWS 中合規的概觀。

文件修訂

若要收到此白皮書更新的通知，請訂閱 RSS 摘要。

update-history-change	update-history-description	update-history-date
已更新	重大更新	2021 年 2 月 15 日
初次出版	白皮書初始出版	2019 年 1 月 3 日

聲明

客戶應負責對本文件中的資訊自行進行獨立評估。本文件：(a) 僅供參考之用，(b) 代表目前的 AWS 產品供應與實務，如有變更恕不另行通知，以及 (c) 不構成 AWS 及其附屬公司、供應商或授權人的任何承諾或保證。AWS 產品或服務以「現況」提供，不提供任何明示或暗示的擔保、主張或條件。AWS 對其客戶之責任與義務，應受 AWS 協議之約束，且本文件並不屬於 AWS 與其客戶間之任何協議的一部分，亦非上述協議之修改。